Computer Programs in Biomedicine 18 (1984) 94–98
Elsevier

CPB 0642A

Appendix A – Programming Manual

Programming manual for IDENT, a parametric and nonparametric linear systems identification package

Susan A.S. Werness and David J. Anderson

Kresge Hearing Research Institute, University of Michigan, Ann Arbor, MI 48109, USA

1, INTRODUCTION

This document is intended to provide helpful information to anyone wishing to adapt IDENT for use on a computer other than an LSI-11, or on an operating system other than RT-11 version 4, or with different data structures.

An overall picture of IDENT is provided by Tables 1–7, in which the common variables and the overlay structure are outlined. The function of each subroutine is noted briefly. The main program establishes parameters for the program run. Part of it is concerned with specific data structures or with RT-11 functions and would probably require change. The subroutine, COMMAN, is called by the main program for each set of input and output data. COMMAN is the calling routine for all of the commands processed by IDENT and thus must be located in the root segment.

Also provided in this manual are linking instructions, a list and description of RT-11 functions used in IDENT. A discussion of the structure of a main program is provided. Also, the means by which the parameter estimator's convergence rate could be altered by some minor program changes is outlined.

2. LINKING INSTRUCTIONS

On the RT-11 version 4 system, the linking instructions are as follows:

```
R. LINK
IDENT,, = ROOTSG/B:2000/P:400
PAREST/0:1/C
NONPAR/0:1/C
RSD/0:1/C
UNIVRT/0:1/C
SING/0:1/C
TEACH/0:1
```

These instructions indicate that the modules, ROOTSG (the main program and COMMAN) comprise the root segment or the portion of the program which remains in memory at all times. The next six modules share the same memory region. Thus, certain restrictions were followed in order that this overlay structure works. A given subroutine's return path must remain in memory. For example, the main program or COMMAN may call any subroutine in the system since ROOTSG is always resident. However, a call cannot be made from PAREST to a routine in RSD since these 2 modules share the same memory area.

The program was linked with 2 options. The /B:2000 option moves the program's starting address 1000 octal words higher. In RT-11 version 4, the double precision routine, DPRQD, needs a larger stack area (the stack is located below the program) than is automatically allocated. /P:400 just expands the linking table.

The program fits in a 28 K memory with some difficulty. The program size is dictated by the size of the root segment (including the COMMON area) and the size of the largest overlay, TEACH. To save space, these modules were compiled with a no linenumbers option which suppressed the inclusion of internal line numbers in the object files. The /B:2000 option aggravates the problem by 1000 words. If necessary, the TEACH overlay could be dropped or could be converted to get the instructions out of a file instead of containing a multitude of format statements.

2. RT-11 FUNCTIONS CALLED IN IDENT

2.1. ICSI

Example of a call:

```
5 IF(ICSI(SPEC,EXT,,,Ø).NE.Ø) GO TO 5
```

Location of calls:
(1) Main program – to set up files with the input and output data
(2) OPPAR – to set up the command file

Explanation:
The ICSI function calls the RT-11 Command String Interpreter in special mode to parse a command string and return file descriptors and options to the program.

SPEC is the 39 word area to receive the file specifications.
The format of this area (considered as a 39-element INTEGER*2 array) is:
SPEC(1)-SPEC(4) – output file 1 specification.
SPEC(5) – output file 1 length.
SPEC(6)-SPEC(9) – output file 2 specification.
SPEC(10) – output file 2 length.
SPEC(11)-SPEC(14) – output file 3 specification.
SPEC(15) – output file 3 length.
SPEC(16)-SPEC(19) – input file 1 specification.
SPEC(20)-SPEC(23) – input file 2 specification.
SPEC(24)-SPEC(27) – input file 3 specification.
SPEC(28)-SPEC(31) – input file 4 specification.
SPEC(32)-SPEC(35) – input file 5 specification.
SPEC(36)-SPEC(39) – input file 6 specification.

EXT is the table of Radix-50 default file types to be assumed when a file is specified without a file type

the Ø indicates that there are no options

2.2. IASIGN

Location of calls:
(1) Main program – to set up files with the input and output data
(2) OPPAR – to set up the command file

Explanation:
The IASIGN function sets information in the FORTRAN logical unit table (overriding the defaults) so that the specified information is used when the FORTRAN Object Time System (OTS) opens the logical unit. This function can be used with ICSI to allow a FORTRAN program to accept a standard CSI input specification. IASIGN must be called before the unit is opened; that is, before any READ, WRITE, PRINT, TYPE, or ACCEPT statements are executed that reference the logical unit.

Form: i = IASIGN (lun, idev[, ifiltyp[, isize[, itype]]])

where: lun is an INTEGER*2 variable, con - stant, or expression specifying the FORTRAN logical unit for which information is being specified.
idev is a one-word Radix-50 device name; this can be the first word on an ICSI input or output file specification.
ifiltyp is a three-word Radix-50 file name and file type; this can be words 2 through 4 of an ICSI input or output file specification.
isize is the length (in blocks) to allocate for

an output file; this can be the fifth word of an ICSI output specification. If 0, the larger of either one-half the largest empty segment or the entire second largest empty segment is allocated. If the value specified for length is -1, the entire largest empty segment is allocated.

itype is an integer value determining the optional attributes to be assigned to the file. This value is obtained by adding the values that correspond to the desired operations.

1	use double buffering for output
2	open the file as a temporary file
4	Force a LOOKUP on an existing file during the first I/O operation (otherwise, the first FORTRAN I/O operation determines how the file is opened). For example, if the next I/O operation is a write, an IENTER is performed on the specified logical unit. A read causes a LOOKUP.
8	expand carriage control information
16	do not expand carriage control information
32	file is read-only

2.3. GETSTR

Example of a call:
CALL GETSTR (5,FORAY,19,IER)

Location of the call:
Main program – to get data format type

Explanation:
The GETSTR subroutine reads a formatted ASCII record from Fortran logical unit 5 (the terminal) into the array: FORAY. The string can be up to 19 characters. IER indicates errors. The data is truncated (trailing blanks removed) and a null byte inserted at the end to form a character string.

2.4. CLOSE

Example of a call:

CALL CLOSE (3)

Location of the call:
(1) Main program – close command file and data files.
(2) OPPAR – closes the command file.
Explanation:
CLOSE simply closes the file associated with the indicated logical unit number.

3. MAIN PROGRAM

The main program may be tailored to a user's specific needs. However, there are certain things that a main program in IDENT should do:

(1) Initialization of the operation parameters.

(2) Initialization of the data sampling rate.

(3) Initialization of the data format type.

(4) Set up files of data to be analyzed.

(5) Read in the data (by calling RSSF).

If the data arrays are a different length than 512 words, then the dimensions of the common arrays PR, CR, WH, PD, and RD may be changed. However, increasing these dimensions will necessitate the program occupying a larger segment of memory.

4. CONTROL OF THE PARAMETER ESTIMATOR

Currently, it is not possible to exert directly any control over the parameter estimator's behavior. However, there are some minor program changes in the parameter estimator, RLSTSQ, which could be made to alter its convergence rate. If one really wanted to play with these parameters, perhaps they could be included in the list of operation parameters. Following are listed these easy to change parameters.

(1) *Convergence criterion*: When parameter estimates obtained after the previous iteration differs from those from the current iteration by less than .01, the algorithm stops. This convergence criterion can be relaxed or tightened. The statement containing the .01 is near the line numbered 9754.

(2) *Exponential weighting window parameters*: An exponential weighting window is applied to the parameter estimator so that more recent data have more 'weight' in the forming of the parameter estimates. The window is of the form:

$$\text{RHO} = (1. - \text{DELRHO}) * \text{RHO} + \text{DELRHO}$$

If RHO is chosen to be close to 1 and DELRHO chosen to be small, RHO will tend to 1 exponentially. RHO is .99. DELRHO is .001 initially.

(3) *Maximum number of iterations*: This number is currently 20. It could be changed in a statement also near line number 9754.

TABLE 2

Subroutines of the NONPAR overlay

Name	Function
NONPAR	Calculation of a nonparametric transfer function and impulse response
INSUB	Calculation of a smoothed univariate spectrum
RESAMP	Resampling subroutine
SMSPEC	Spectral smoothing subroutine
CFFT	FFT algorithm
MCACB	Moves a complex array to another complex array
STRAR	Increases the size of a complex array by internal padding with zeros
COMAR	Decreases the size of a complex array by taking out the center
WINAR2	Window with 2 coefficients
MUCXCC	Multiplication of a complex array by a complex conjugate
DICXC	Division of a complex array by another

TABLE 1

Common area

Variable name	Dimension	Type	Use
RD	512	Real	Scaled output array
PD	512	Real	Scaled input assay
CR	512	Real	Differenced or detrended output
PR	512	Real	Differenced or detrended input used for residual
WH	512	Real	Prewhitened input
PAR	50	Real	Parameter estimates
NTFP	1	Integer	Number of transfer function poles
MTFZ	1	Integer	Number of transfer function zeros
Noise	1	Integer	Number of noise poles
Noise	1	Integer	Number of noise zeros
NN1	1	Integer	Beginning point of input, output arrays
NN2	1	Integer	Endpoint of input, output arrays
IDIF	1	Integer	Input array difference parameter
IDIF2	1	Integer	Output array difference parameter
MAX	1	Integer	Increment to NN1 for residual calculation
NPTS	1	Integer	Maximum number of points in a data array

TABLE 3

Subroutines of the PAREST overlay

Name	Function
RLSTSQ	Main parameter estimation algorithm
FILL	Calculation of observation matrix and linearized observation matrix
LIN	Filtering subroutine used in linearization
DFILT	Filtering subroutine used in linearization
SHIFT	Shifts output array to the right
MULTI	Multiplication of a square $N \times N$ matrix by an N column vector
MULT2	Multiplication of an N row vector by an $N \times N$ square matrix
MULT3	Multiplication of an N row vector by an N column vector to yield a scalar
MULT4	Multiplication of an N column vector by an N row vector to yield an $N \times N$ square matrix
MULT5	Multiplication of 2 square matrices
VADD1	Addition of 2 column vectors
VADD2	Addition of 2 square matrices
SCMLT1	Multiplication of a vector by a scalar
SCMLT2	Multiplication of a square matrix by a scalar

TABLE 4

Subroutines of the RSD overlay

Name	Function
RSD	Residual estimation
FILL2	Calculation of the observation matrix used in residual estimation
MULT3B	Multiplication of an N row vector by an N column vector to yield a scalar
EV	Calculation of the parametric spectrum, the noise variance, signal variance, and signal to noise ratio
RSSF	Reads and scales data
HEADRD	Reads header data
GETCOM	Gets a command from the user or a command file
IDINIT	Initializes some of the common variables

TABLE 5

Subroutines of the UNIVRT overlay

Name	Function
IDNOIS	Calling routine for univariate (or bivariate) statistics including normalized autocovariance or cross covariance, mean, variance, partial autocorrelation, Q and S statistics
DECPLT	DECWRITER plotting routine
XCOR	Cross or autocovariance lags
PACF	Partial autocorrelation lags
MEANS	Mean and variance
DIFF	Differences an array
RGRSSN	Linear regression
CASDIF	Detrends and/or differences an array
DPCR	Sets data endpoints and difference parameters
OPPAR	Changes operation parameters

TABLE 6

Subroutines of the SING overlay

Name	Function
ROOTF	Setting up and calling routine for the root finder
DPRQD	Double precision root finder
MAGANG	Changes real and imaginary part format of the complex roots into radius and frequency format

TABLE 7

Subroutines of the TEACH overlay

Name	Function
TEACH	Finds out which command to explain
IDINST	Contains explanations for all the commands
CLRSCR	Does several form feeds